

Análisis del retardo de extremo a extremo

Práctica 1

1. Introducción

Debido a que el mecanismo de transmisión del protocolo IP (Internet Protocol), denominado *best-effort*, no ofrece calidad de servicio (QoS: Quality of Service), los paquetes pueden perderse, duplicarse, llegar en desorden o retardarse excesivamente. Las aplicaciones multimedia en Internet utilizan memorias de reproducción (*playout*) en el receptor para suavizar la variabilidad del retardo de extremo a extremo. Este tipo de procedimientos se usan tanto en aplicaciones de audio como en aplicaciones de vídeo.

Una vez suavizada la variabilidad del retardo, los paquetes se reproducen en el receptor a un tiempo programado por el algoritmo de *playout*. Los paquetes que llegan después de su tiempo programado de reproducción se consideran perdidos y no se reproducen. Esto hace que la tasa de pérdidas total en una aplicación multimedia se incremente debido a pérdidas locales en la memoria del receptor. Este tipo de algoritmos implican un compromiso (*tradeoff* en inglés) entre interactividad y retraso: mientras más espere el receptor a que lleguen paquetes, más paquetes llegarán pero la interactividad se verá disminuida. El análisis del retardo de extremo a extremo (end-to-end delay, o bien OWD por sus siglas en inglés *One Way Delay*) es entonces de gran importancia no sólo para aplicaciones multimedia, sino para varios tipos más de aplicaciones en Internet.

El retardo de extremo a extremo (OWD), es el tiempo medido desde la transmisión de un paquete hasta su recepción en el otro extremo. En esta práctica, usted descargará seis trazas OWD que la Profra. Sue Moon ha puesto disponibles en su sitio web.

2. Procedimiento

La experimentación a realizar, permitirá estudiar el comportamiento del retardo de extremo a extremo en Internet con la ayuda de trazas. Dichas trazas permitirían también implementar simulación basada en ellas. La simulación basada en trazas es un tipo de simulación que permite la evaluación de algoritmos de manera tal que los resultados serían idénticos a una implementación real. Este tipo de simulación requiere contar con trazas obtenidas a partir de una aplicación. Posteriormente, estas trazas se inyectan ya sea en simuladores que implementan el/los algoritmo(s) a evaluar, o bien directamente en un modelo analítico. De esta manera, es posible evaluar cualquier mecanismo antes de

implementarlo en una aplicación real. La desventaja que presenta la simulación guiada por trazas, es la dificultad para obtenerlas.

NeVoT es una aplicación de audio que fue desarrollada por el Prof. Henning Schulzrinne y usada en los inicios de la era de voz sobre IP (VoIP). Ofrecía soporte para varios codificadores de audio, RTP/RTCP, unicast y multicast. NeVoT permitía guardar las trazas de una conversación de audio en un formato apropiado. Información adicional sobre NeVoT está disponible en <http://www.cs.columbia.edu/~hgs/rtp/nevot.html>.

Se hace mención de NeVoT, para referirnos a la herramienta que permitió a la Profra. Moon obtener las trazas usadas en esta práctica. Usted no necesitará ejecutar NeVoT, sólo deberá descargar las trazas y utilizarlas para realizar la experimentación requerida.

2.1. Identificación de las trazas de audio

Las trazas que usaremos están disponibles en el sitio de Sue Moon y fueron utilizadas en [1]. Numerosos trabajos han utilizado también estas mismas trazas para estudiar el retardo de extremo a extremo y su caracterización, principalmente en aplicaciones de VoIP (*Voice over IP*).

Una vez que descargue usted las trazas, observará que todas ellas siguen un formato de tres columnas. La primera columna contiene una "D" si el evento registrado es un paquete recibido y contiene un "!" si es un período de silencio. Las dos columnas restantes son estampas de tiempo virtuales vat (<http://www-nrg.ee.lbl.gov/vat/>). En la segunda columna, está la estampa de tiempo del receptor y en la tercera columna la estampa de tiempo del emisor. En las trazas, el audio está codificado en PCM de 8KHz. Un paquete se genera cada 20 ms en una frase. Un extracto de la traza 01 se muestra a continuación:

```
D 131282239 131921839
D 131282239 131921999
D 131282239 131922159
D 131282239 131922319
D 131283263 131922479
! 131922959
D 131284287 131922959
D 131284287 131923119
D 131284287 131923279
```

Como se puede observar, en este ejemplo se ilustra parte de una ráfaga de paquetes, todos con la primera columna en "D" para indicar que el i -ésimo paquete fue recibido en el receptor con estampa de tiempo t_i^r indicada en la segunda columna y transmitido con estampa de tiempo t_i^t , en la tercera columna. El registro que contiene un "!" indica que en ese instante comienza un período de silencio. Puesto que se trata de trazas de voz sobre IP, se debe observar que cuando hablamos no lo hacemos continuamente;

en cambio, realizamos pausas para estructurar mejor nuestro discurso. Entonces, en una sesión de VoIP se producen paquetes cuando el emisor habla y no se transmite ningún paquete al hacer pausas. Esto distingue los registros con “D” y con “!” en las trazas de retardo que utilizamos en esta práctica.

Una simple diferencia, $t_i^r - t_i^t$, entre esas estampas de tiempo no bastaría para caracterizar básicamente al retardo de extremo a extremo. Esto es porque los relojes del emisor y del receptor no están sincronizados y, por ende, una simple diferencia entre las estampas de tiempo del receptor y del transmisor podría devolver un retardo negativo, lo que sería físicamente irrealizable; eso querría decir que *el paquete i llegó al receptor antes de haber sido enviado (!!!)*. Como la intención es caracterizar el retardo de extremo a extremo; es decir, observar su comportamiento, la Profra. Sue Moon propone simplemente añadir un elemento más a la sustracción; dicho elemento es la mínima diferencia de retardo encontrada en toda la traza. Esto significa encontrar $t_{\min} = \min\{t_i^r - t_i^t\}$, $\forall i$. Como las estampas de tiempo están codificadas con el protocolo RTP y han sido obtenidas con voz muestreada a 8000 Hz, finalmente podremos observar el comportamiento del retardo de extremo a extremo del paquete i con:

$$d_{\text{end-to-end}}^i = \frac{t_i^r - t_i^t - t_{\min}}{8000} [\text{seg}]. \quad (1)$$

Para encontrar el *tiempo de sesión* para cada paquete, el análisis es idéntico al de una llamada telefónica tradicional, ya sea fija o celular: *el que llama, paga*. Entonces, el tiempo de sesión estará basado en el reloj del transmisor. Este tiempo comenzará a correr a partir de la primera estampa de tiempo t_1^t , hasta la última t_N^t , dado que N es el número total de paquetes en la sesión. Entonces:

$$t_{\text{session}}^i = \frac{t_i^t - t_1^t}{8000} [\text{seg}]. \quad (2)$$

Para analizar el retardo de extremo a extremo, entonces, usted deberá realizar las gráficas de las Ecuaciones (1) y (2) para cada una de las seis trazas, a gran escala y a pequeña escala, como se indica más abajo.

2.2. Manipulación de trazas con AWK

Como hemos visto en las actividades técnicas previas a esta práctica, AWK es un lenguaje de procesamiento de texto que se utiliza frecuentemente para manipular texto formateado. Como las trazas de retardo de extremo a extremo que usamos en esta práctica están organizadas en columnas, AWK es una excelente herramienta para el análisis que vamos a realizar.

Para analizar algunas estadísticas básicas de las sesiones de VoIP capturadas en las trazas, realice lo siguiente:

1. Escriba un script AWK para contar el número total de frases (*talkspurts*) en una traza de retardo.
2. Realice lo mismo, pero ahora para contar el número total de paquetes que llegaron al receptor.
3. Escriba un script AWK para encontrar la diferencia mínima entre la estampa de tiempo del receptor menos de la del emisor.
4. Escriba un script AWK que reciba una traza como entrada y entregue un archivo con dos columnas: tiempo de la sesión en segundos, y retardo de extremo a extremo en segundos.
5. Escriba un script AWK que calcule el retardo promedio de extremo a extremo en una sesión.
6. Comente el código de cada script AWK. Importante: usted debe ser el autor del código que está reportando.

2.3. Comportamiento del retardo de extremo a extremo

1. Utilice los resultados obtenidos en el ejercicio anterior para trazar el retardo de extremo a extremo de cada una de las seis sesiones de VoIP. Utilice alguna de las herramientas empleadas en las actividades técnicas previas para tal fin; e.g., GNUPlot, Octave o Matlab. Para trazar gráficas, evite usar hojas de cálculo.

Se le solicita obtener dos tipos distintos de gráficas por cada traza: (a) *Gráfica a gran escala*, en donde trazará todos los elementos del retardo de extremo a extremo. Esto será útil para observar el comportamiento macroscópico del retardo en una sesión de VoIP. Por otro lado, trace igualmente una (b) *gráfica a pequeña escala*, para analizar el comportamiento microscópico del retardo; e.g., una gráfica que cubra dos o tres frases en la conversación. Para el caso (b) se sugiere usar gráficas con líneas verticales (stem o impluses, para Octave y GNUPlot, respectivamente).

2. ¿Es posible observar fenómenos particulares en las gráficas que acaba de obtener? Comente al respecto.
3. En una tabla, incorpore las estadísticas obtenidas con sus scripts; i.e., número de traza, total de paquetes, total de frases, retardo promedio de extremo a extremo.
4. Investigue al menos dos métodos propuestos en la literatura para eliminar la desviación de la pendiente en el retardo. Describa brevemente su principio de funcionamiento.

Referencias

- [1] Sue Moon, Jim Kurose, and Don Towsley. Packet audio playout delay adjustment: Performance bounds and algorithms. *ACM/Springer Multimedia Systems*, 6(1), February 1998. <http://an.kaist.ac.kr/~sbmoon/paper/intl-journal/1998-acm-multimedia-voip.pdf>.